



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/726,308

12/01/2003

Neal M. Gafter

SUN-P9044

3084

57960

7590

09/19/2006

SUN MICROSYSTEMS INC.
C/O PARK, VAUGHAN & FLEMING LLP
2820 FIFTH STREET
DAVIS, CA 95618-7759

EXAMINER

CHEN, QING

ART UNIT

PAPER NUMBER

2191

DATE MAILED: 09/19/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/726,308	Applicant(s) GAFTER ET AL.	
	Examiner Qing Chen	Art Unit 2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 07 August 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-14 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-14 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This Office action is responsive to the amendments filed on August 7, 2006. **Claims 1 and 8** have been amended. **Claims 15-21** have been cancelled. **Claims 1-14** are currently pending and have been considered below.

Response to Amendment

2. Applicant's amendments to the specification appropriately address the objection to the disclosure due to informalities. Accordingly, this objection is withdrawn in view of Applicant's amendments.

3. Applicant's amendments to Claim 8 do not appropriately address the rejection of Claims 8-14 under 35 U.S.C. § 101, based on non-statutory subject matter. Accordingly, this rejection is maintained and further explained below.

Response to Arguments

4. Applicant's arguments filed on August 7, 2006 have been fully considered, but they are not persuasive.

In response to Applicant's arguments regarding Claims 1 and 8, the Examiner maintains that Thatte et al. disclose registering a unique factory to build each component.

Thatte et al. disclose that in response to a call to request creation of a component by the client program, the "CoCreateInstance()" API function first loads the class' executable file after

looking up the registry entry to identify the executable class for the file. Then, the “CoCreateInstance()” API function uses the class factory in the executable file to create an instance of the COM object (*see Column 10, Lines 41-47*). Thatte et al. further disclose that classes of COM objects are uniquely associated with class identifiers (“CLSIDs”), and registered by their CLSID in a system configuration database referred to as the “registry.” The registry entry for a COM object class associates the CLSID of the class with information identifying an executable file that provides the class (*e.g.*, a DLL file having a class factory to produce an instance of the class) (*see Column 10, Lines 24-31*). The class factory in the executable file is the disclosed “factory” as required by the claims, not the “CoCreateInstance()” API function.

Claim Rejections - 35 USC § 101

5. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

6. **Claims 8-14** are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 8-14 recite computer-readable storage medium as a claimed element. The preamble of independent Claim 8 recites that the computer-readable storage medium includes various storage devices (*e.g.*, magnetic and optical storage devices, disk drives, CDs, and DVDs). However, the term “include” is inclusive or open-ended and does not exclude additional, unrecited elements. In addition, the term “include” does not preclude computer-readable storage

Art Unit: 2194

medium from encompassing computer instruction signals embodied in a transmission medium as disclosed in page 6, paragraph [0024] of the specification. Claims that recite nothing but the physical characteristics of a form of energy, such as a frequency, voltage, or the strength of a magnetic field, define energy or magnetism *per se*, and as such are nonstatutory natural phenomena. *O'Reilly v. Morse*, 56 U.S. (15 How.) 62, 112-14 (1853). Moreover, it does not appear that a claim reciting a signal encoded with functional descriptive material falls within any of the categories of patentable subject matter set forth in § 101.

Claim Rejections - 35 USC § 102

7. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

8. **Claims 1-6 and 8-13** are rejected under 35 U.S.C. 102(b) as being anticipated by **Thatte et al.** (US 6,442,620).

As per **Claim 1**, Thatte et al. disclose a method for facilitating typesafe software design while supporting structured composition of a software system, comprising:

A. Receiving a first invocation of the software system (*see Column 9, Lines 15-17; and Column 10, Lines 16-20*);

B. Assigning a first context to the first invocation (*see Column 12, Lines 8-14*);

C. Examining the first invocation to locate components of the first invocation (*see Column 12, Lines 40-46*);

D. Registering a unique factory to build each component, wherein these factories are registered using the first context (*see Column 10, Lines 38-44*); and

E. When a component is needed, building the component using the unique factory associated with the component, whereby building the component after each component has a registered factory eliminates potential problems with initialization circularity (*see Column 10, Lines 44-49*).

As per **Claim 2**, Thatte et al. disclose a method for facilitating typesafe software design while supporting structured composition of a software system as in **Claim 1 above**, and further disclose that the method comprising:

A. Receiving a second invocation of the software system (*see Column 9, Lines 15-17; and Column 10, Lines 16-20*);

B. Assigning a second context to the second invocation (*see Column 12, Lines 8-14*);

C. Examining the second invocation to locate components of the second invocation (*see Column 12, Lines 40-46*);

D. Registering a unique factory to build each component, wherein these factories are registered using the second context (*see Column 10, Lines 38-44*); and

E. When a component is needed, building the component using a factory associated with the component, whereby building the component after each component has a registered factory eliminates problems with initialization circularity (*see Column 10, Lines 44-49*).

As per **Claim 3**, Thatte et al. disclose a method for facilitating typesafe software design while supporting structured composition of a software system **as in Claim 2 above**, and further disclose that the components from the second invocation are not available to the first invocation (*see Column 13, Lines 31-33 and 53-58*).

As per **Claim 4**, Thatte et al. disclose a method for facilitating typesafe software design while supporting structured composition of a software system **as in Claim 1 above**, and further disclose that the method comprising providing an additional factory for an extended component of the first invocation (*see Column 11, Lines 9-22*).

As per **Claim 5**, Thatte et al. disclose a method for facilitating typesafe software design while supporting structured composition of a software system **as in Claim 1 above**, and further disclose that registering the unique factory to build each component involves placing a key and a related factory identifier into a storage structure (*see Column 10, Lines 24-27*).

As per **Claim 6**, Thatte et al. disclose a method for facilitating typesafe software design while supporting structured composition of a software system **as in Claim 5 above**, and further disclose that building the component using the factory associated with the component involves using the key to retrieve the related factory identifier from the storage structure (*see Column 10, Lines 27-31*).

As per **Claim 8**, Thatte et al. disclose a computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for facilitating typesafe software design while supporting structured composition of a software system, wherein the computer-readable storage medium includes magnetic and optical storage devices, disk drives, magnetic tape, CDs (compact discs), and DVDs (digital versatile discs or digital video discs), the method comprising:

- A. Receiving a first invocation of the software system (*see Column 9, Lines 15-17; and Column 10, Lines 16-20*);
- B. Assigning a first context to the first invocation (*see Column 12, Lines 8-14*);
- C. Examining the first invocation to locate components of the first invocation (*see Column 12, Lines 40-46*);
- D. Registering a unique factory to build each component, wherein these factories are registered using the first context (*see Column 10, Lines 38-44*); and
- E. When a component is needed, building the component using the unique factory associated with the component, whereby building the component after each component has a registered factory eliminates potential problems with initialization circularity (*see Column 10, Lines 44-49*).

As per **Claim 9**, Thatte et al. disclose a computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for facilitating typesafe software design while supporting structured composition of a software system as in **Claim 8** above, and further disclose that the method comprising:

Art Unit: 2194

- A. Receiving a second invocation of the software system (*see Column 9, Lines 15-17; and Column 10, Lines 16-20*);
- B. Assigning a second context to the second invocation (*see Column 12, Lines 8-14*);
- C. Examining the second invocation to locate components of the second invocation (*see Column 12, Lines 40-46*);
- D. Registering a unique factory to build each component, wherein these factories are registered using the second context (*see Column 10, Lines 38-44*); and
- E. When a component is needed, building the component using a factory associated with the component, whereby building the component after each component has a registered factory eliminates problems with initialization circularity (*see Column 10, Lines 44-49*).

As per **Claim 10**, Thatte et al. disclose a computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for facilitating typesafe software design while supporting structured composition of a software system **as in Claim 9 above**, and further disclose that the components from the second invocation are not available to the first invocation (*see Column 13, Lines 31-33 and 53-58*).

As per **Claim 11**, Thatte et al. disclose a computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for facilitating typesafe software design while supporting structured composition of a software system **as in Claim 8 above**, and further disclose that the method comprising providing an additional factory for an extended component of the first invocation (*see Column 11, Lines 9-22*).

As per **Claim 12**, Thatte et al. disclose a computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for facilitating typesafe software design while supporting structured composition of a software system **as in Claim 8 above**, and further disclose that registering the unique factory to build each component involves placing a key and a related factory identifier into a storage structure (*see Column 10, Lines 24-27*).

As per **Claim 13**, Thatte et al. disclose a computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for facilitating typesafe software design while supporting structured composition of a software system **as in Claim 12 above**, and further disclose that building the component using the factory associated with the component involves using the key to retrieve the related factory identifier from the storage structure (*see Column 10, Lines 27-31*).

Claim Rejections - 35 USC § 103

9. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

10. **Claims 7 and 14** are rejected under 35 U.S.C. 103(a) as being unpatentable over **Thatte et al.** (US 6,442,620) in view of **Gibbons et al.** (US 5,761,511).

As per **Claim 7**, Thatte et al. disclose a method for facilitating typesafe software design while supporting structured composition of a software system **as in Claim 6 above**. However, Thatte et al. does not explicitly disclose that the storage structure includes a hash table.

In the same field of endeavor, Gibbons et al. discloses a system and method for extending the functionality of data structures, where the data structures are made up of connected collections of objects, and mechanisms for projecting such data structures of objects from one type space to another. In the system, a shadow map is implemented by calling upon a factory object and the results are cached in a hash table (*see Column 9, Lines 33-35*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize a hash table as a storage structure in the system of Thatte et al., since hash table is a very common type of lookup structure and can be used to implement a configuration database. One would have been motivated to utilize a hash table as a storage structure in order to provide a constant lookup time on average, regardless of the number of items in the hash table.

As per **Claim 14**, Thatte et al. disclose a computer-readable storage medium storing instructions that when executed by a computer cause the computer to perform a method for facilitating typesafe software design while supporting structured composition of a software system **as in Claim 13 above**. However, Thatte et al. does not explicitly disclose that the storage structure includes a hash table.

In the same field of endeavor, Gibbons et al. discloses a system and method for extending the functionality of data structures, where the data structures are made up of connected

Art Unit: 2194

collections of objects, and mechanisms for projecting such data structures of objects from one type space to another. In the system, a shadow map is implemented by calling upon a factory object and the results are cached in a hash table (*see Column 9, Lines 33-35*).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize a hash table as a storage structure in the system of Thatte et al., since hash table is a very common type of lookup structure and can be used to implement a configuration database. One would have been motivated to utilize a hash table as a storage structure in order to provide a constant lookup time on average, regardless of the number of items in the hash table.

Conclusion

11. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

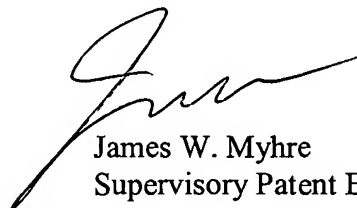
Art Unit: 2194

Any inquiry concerning this communication or earlier communications from the Examiner should be directed to Qing Chen whose telephone number is 571-270-1071. The Examiner can normally be reached on Monday through Thursday from 7:30 AM to 4:00 PM. The Examiner can also be reached on alternate Fridays.

If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, James W. Myhre, can be reached on 571-270-1065. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

QC / *QC*
September 13, 2006


James W. Myhre
Supervisory Patent Examiner